Nonlinear Feature Extraction using Multilayer Perceptron based Alternating Regression for Classification and Multiple-output Regression Problems

Ozde Tiryaki[1,2] and C. Okan Sakar[1]
[1]*Department of Computer Engineering, Bahcesehir University, Istanbul, Turkey*
[2]*NETAS Telecommunication Company, Kurtkoy, Istanbul, Turkey*

Keywords: Alternating Regression (Ar), Multiple-output Regression, Neural Networks, Kernel Canonical Correlation Analysis (Kcca), Nonlinear Dimensionality Reduction.

Abstract: Canonical Correlation Analysis (CCA) is a data analysis technique used to extract correlated features between two sets of variables. An important limitation of CCA is that it is a linear technique that cannot capture nonlinear relations in complex situations. To address this limitation, Kernel CCA (KCCA) has been proposed which is capable of identifying the nonlinear relations with the use of kernel trick. However, it has been shown that KCCA tends to overfit to the training set without proper regularization. Besides, KCCA is an unsupervised technique which does not utilize class labels for feature extraction. In this paper, we propose the nonlinear version of the discriminative alternating regression (D-AR) method to address these problems. While in linear D-AR two neural networks each with a linear bottleneck hidden layer are combined using alternating regression approach, the modified version of the linear D-AR proposed in this study has a nonlinear activation function in the hidden layers of the alternating multilayer perceptrons (MLP). Experimental results on a classification and a multiple-output regression problem with sigmoid and hyperbolic tangent activation functions show that features found by nonlinear D-AR from training examples accomplish significantly higher accuracy on test set than that of KCCA.

1 INTRODUCTION

Canonical correlation analysis (CCA) (Hotelling, 1992) is a multivariate statistical analysis technique used to explore and measure the relations between two multidimensional variables. In data analysis, under the presence of two different input representations of the same data or two data sources providing samples about the same underlying phenomenon, CCA is used as an unsupervised feature extraction technique. It aims at finding a pair of linear transformations such that the transformed variables in the lower dimensional space are maximally correlated.

An important limitation of CCA is that it cannot explore the complex relationships between the sets of variables because of its linearity. To address this problem, kernel CCA was proposed (Akaho, 2001; Melzer et al., 2001; Bach and Jordan, 2003) which offers an alternative solution using a method known as the kernel trick (Schölkopf, 2000). The main idea of KCCA is to map the original low-dimensional input space to a high-dimensional feature space using a nonlinear kernel function and then apply CCA in the transformed space. Kernel CCA is capable of detecting nonlinear relationships under the presence of complex situations. KCCA has been used in a broad range of disciplines like biology, neurology, content-based image retrieval and natural language processing (Huang et al., 2009; Li and Shawe-Taylor, 2006; Sun and Chen, 2007; Cai and Huang, 2017; Chen et al., 2012).

Another important limitation of CCA and KCCA is that under the presence of class labels in supervised learning problems, they do not utilize the class labels for feature extraction but only target to find the maximally correlated covariates of both views. Therefore, covariates explored by these unsupervised methods preserve the correlated information at the expense of losing the important discriminative information which can be helpful in separating class examples from each other.

In this paper, we propose the nonlinear version of the discriminative alternating regression (D-AR) network (Sakar and Kursun, 2017) which is based on

Tiryaki, O. and Sakar, C.
Nonlinear Feature Extraction using Multilayer Perceptron based Alternating Regression for Classification and Multiple-output Regression Problems.
DOI: 10.5220/0006848901070117
In *Proceedings of the 7th International Conference on Data Science, Technology and Applications (DATA 2018)*, pages 107-117
ISBN: 978-989-758-318-6

the alternating regression (AR) method (Sakar et al., 2014b). The AR approach is first described in (Wold, 1966) and its neural network adaptations have been applied in (Lai and Fyfe, 1998), (Pezeshki et al., 2003) and (Hsieh, 2000) to extract robust CCA covariates. In the previously proposed linear D-AR (Sakar and Kursun, 2017; Sakar et al., 2014b), two neural networks each with a linear bottleneck hidden layer are trained to learn both class labels and covariate outputs using alternating regression approach. Having both class labels and covariate outputs in the output layer improves the discriminative power of the extracted features. Besides, feature extraction without the use of sensitive sample covariance matrices makes the network more robust to outliers (Sakar and Kursun, 2017). The non-linear version of D-AR has a nonlinear activation function in the hidden layers of the alternating multilayer perceptrons (MLP). Covariate outputs are alternated between the corresponding MLPs in order to maximize the correlation between two views. In our experiments, we compare the classification and regression performance of the features extracted by the proposed nonlinear D-AR with that of linear D-AR, CCA, and KCCA on publicly available emotion recognition and residential building datasets. We use two nonlinear activation functions, sigmoid and hyperbolic tangent, in the hidden layer of nonlinear D-AR and present the results for different training set sizes and number of covariate outputs.

The rest of this paper is structured as follows. In Section II, we give brief information on the datasets used; emotion recognition and residential building. Section III provides background on CCA, KCCA, MLP, and linear D-AR. In Section IV, we present the details of the proposed nonlinear D-AR method. Experimental results are given in Section V. The conclusions are given in Section VI.

2 DATASET

The Cohn-Kanade (CK+) facial expression database (Lucey et al., 2010) is a commonly used benchmarking dataset in emotion recognition tasks. This dataset consists of 320 video clips recorded from 118 subjects, each categorized with an emotion label. Each video clip in this dataset belongs to one of the seven emotions which are anger, contempt, disgust, fear, happiness, sadness, and surprise. The samples in this dataset can be represented using different feature extraction techniques. In our experimental study, the first view consists of appearance-based features (Sakar et al., 2014a; Karaali, 2012; Sakar et al., 2012) which are obtained using the difference between the first frame of the video clip (the neutral facial expression) and the corresponding last frame (the peak frame of the emotion). Each sample in this representation has 4096 (64×64) features (pixels). The second view consists of the geometric set of features (Sakar et al., 2014a; Ulukaya, 2011; Karaali, 2012), which are constituted by subtracting the coordinates of landmark points of the neutral face expression from the coordinates of the landmark points of the target expression. The feature vector in the second view consists of 134 features obtained from 67 landmark points, each of which represented with x and y coordinates.

The Residential Building dataset (Rafiei and Adeli, 2015) is one of the most recent regression datasets in UCI Machine Learning Repository (Asuncion and Newman, 2007). The dataset consists of 372 instances with 31 features which are collected under 2 different views. While the first view containing physical and financial values belonging to the project has 12 features, the second view containing general economic variables and indices consists of 19 features. Residential building dataset is a multiple output regression problem that contains two output variables which are construction costs and sale prices of single-family residential apartments. In this study, we construct a single non-linear D-AR network that predicts both of these outputs during the feature extraction step.

3 METHODS

3.1 CCA

Canonical correlation analysis (CCA) (Hotelling, 1992) is a way of measuring the linear relationship between two multidimensional views that are related with each other. Given two datasets X (N×m) and Y (N×n)

$$\begin{aligned} X &= [x_1 \quad x_2 \quad x_3 \quad \cdots \quad x_N] \\ Y &= [y_1 \quad y_2 \quad y_3 \quad \cdots \quad y_N] \end{aligned} \tag{1}$$

where N is the total number of the instances, m and n are the number of features in datasets X and Y respectively, CCA aims to find two sets of basis vectors, one for the first view X and the other for the second view Y, such that the correlations between the projections of the variables onto these basis vectors are mutually maximized. More formally, CCA aims tohttps://www.sharelatex.com/project/5a3293785c827c59c12b54c7 maximize the correlation between the linear combinations $w_x^T X$ and $w_y^T Y$:

$$\rho = max_{w_x, w_y} corr(w_x^T X, w_y^T Y) \tag{2}$$

$$\rho = max_{w_x,w_y} \frac{E[(w_x^T X)(w_y^T Y)^T]}{\sqrt{E[(w_x^T X)(w_x^T X)^T]E[(w_y^T Y)(w_y^T Y)^T]}} = max_{w_x,w_y} \frac{w_x^T E[XY^T])w_y}{\sqrt{w_x^T E[XX^T]w_x w_y^T E[YY^T]w_y}} \tag{3}$$

where E denotes the expectation. The total covariance matrix C of (X,Y)

$$C = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} = E\left[\begin{pmatrix} X \\ Y \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}^T \right] \tag{4}$$

is a block matrix, where C_{xx} and C_{yy} are within-set covariance matrices, $C_{xy} = C_{yx}^T$ are betweenset covariance matrices. We can define the equation in (4) as

$$\begin{aligned} \rho &= max_{w_x,w_y} \frac{w_x^T C_{xy} w_y}{\sqrt{w_x^T C_{xx} w_x \cdot w_y^T C_{yy} w_y}} \\ w.r.t \quad & w_x^T C_{xx} w_x = 1 \\ & w_y^T C_{yy} w_Y = 1 \end{aligned} \tag{5}$$

Using the Lagrangian relaxation method, the CCA optimization problem given in (5) is reduced to an eigenvalue problem in the form of $Ax = \lambda Bx$.

$$\begin{aligned} C_{xy} C_{YY}^{-1} C_{yx} w_x &= \lambda^2 C_{xx} w_x \\ C_{yx} C_{XX}^{-1} C_{xy} w_y &= \lambda^2 C_{yy} w_y \end{aligned} \tag{6}$$

The canonical correlations between X and Y can be found by solving the eigenvalue equations

$$\begin{aligned} C_{xx}^{-1} C_{xy} C_{yy}^{-1} C_{yx} w_x &= \lambda^2 w_x \\ C_{yy}^{-1} C_{yx} C_{xx}^{-1} C_{xy} w_y &= \lambda^2 w_y \end{aligned} \tag{7}$$

where the eigenvalues λ are the canonical correlations, the eigenvectors w_x and w_y are the normalized canonical correlation basis vectors. The number of non-zero solutions to these equations are limited to the smallest dimensionality of X and Y. The projections of X and Y onto these canonical vectors, $w_x^T X$ and $w_y^T Y$, are called canonical variables or covariates.

3.2 Kernel CCA

CCA is limited to discovering linear relationships since it maximizes the correlations between linear combinations of the views. To address this problem, kernelized version of CCA called Kernel canonical correlation analysis (KCCA) has been proposed which is capable of identifying the nonlinear relationships between the views (Akaho, 2001). KCCA offers an alternative solution by using a method known as the kernel trick to find nonlinear correlated projections. In KCCA, before performing CCA, first each view is projected into a higher dimensional feature space using a nonlinear kernel function, where the data can be linearly separable. In this stage, KCCA maps x_i and y_i to $\phi(x_i)$ and $\phi(y_i)$

$$\begin{aligned} x = (x_1,\ldots,x_n) &\mapsto S_x = (\phi_1(x),\ldots,\phi_N(x)) \\ y = (y_1,\ldots,y_n) &\mapsto S_y = (\phi_1(y),\ldots,\phi_N(y)). \end{aligned} \tag{8}$$

Then, CCA is applied to the obtained representations $\phi(x_i)$ and $\phi(y_i)$.

Using the definition of the covariance matrix in equation (4), we can rewrite the within-set and between-set covariance matrices, C_{xx} and C_{xy}, as

$$\begin{aligned} C_{xx} &= S_x^T S_x \\ C_{xy} &= S_x^T S_Y \end{aligned} \tag{9}$$

w_x and w_y are the projections of the data onto the directions α and β

$$\begin{aligned} w_x &= S_x^T \alpha \\ w_y &= S_y^T \beta \end{aligned} \tag{10}$$

Substituting into equation (5), we obtain the following

$$\rho = max_{\alpha,\beta} \frac{\alpha^T S_x S_x^T S_y S_y^T \beta}{\sqrt{\alpha^T S_x S_x^T S_x S_x^T \alpha \cdot \beta^T S_y S_y^T S_y S_y^T \beta}} \tag{11}$$

Let $K_x = S_x S_x^T$ and $K_y = S_y S_y^T$ be the kernel matrices, ρ becomes

$$\begin{aligned} \rho &= max_{\alpha,\beta} \frac{\alpha^T K_x K_y \beta}{\sqrt{\alpha^T K_x^2 \alpha \cdot \beta^T K_y^2 \beta}} \\ w.r.t \quad & \alpha^T K_x^2 \alpha = 1 \\ & \beta^T K_y^2 \beta = 1 \end{aligned} \tag{12}$$

In order to resolve computational issues in this high dimensional dataset, partial Gram-Schmidt orthogonolisation (PGSO) is used to approximate the kernel matrices. α and β can be found by resolving

$$\begin{aligned} (K_x kI)^{-1} K_y (K_y kI)^{-1} K_x \alpha &= \lambda^2 \alpha \\ \beta &= \frac{(K_y kI)^{-1} K_x \alpha}{\lambda} \end{aligned} \tag{13}$$

where k is the regularization parameter. Similar to CCA, KCCA is known to be sensitive to outliers (Sakar et al., 2014a; Branco et al., 2005) while deriving the nonlinear correlation subspace. Another important problem of KCCA is its poor generalization ability on unseen test examples (Biemann et al., 2010; Yeh et al., 2014). The previous studies showed

that KCCA tends to overfit to the training set without proper regularization such as using reduced kernel technique (Lee and Huang, 2007; Yeh et al., 2014). In our experiments, since we do not apply a regularization for the proposed MLP based alternating regression technique such as weight decay, not to favor KCCA with an advanced regularization step we apply principal component analysis (PCA) as a preprocessing step to the views and then apply KCCA to the obtained PCA representations (Zhu et al., 2012; He et al., 2005).

3.3 Multilayer Perceptron

The proposed nonlinear D-AR method is based on the use of two alternating multilayer perceptrons. A multilayer perceptron (MLP) is a type of feed-forward artificial neural networks that generates a set of outputs from a set of inputs (Rumelhart et al., 1988). The MLP architecture consists of several layers of nodes between the input and output layers. An activation function is applied to the output of a neuron for decision making. The neuron can learn linear or nonlinear decision boundaries based on the nonlinear activation function of the hidden layer. The most commonly used activation functions are sigmoid, hyperbolic tangent (tanh) and rectified linear unit (ReLU) functions. While the sigmoid function maps the input to the range of 0 to 1 , tanh maps to values between -1 and 1. ReLU allows only positive values to pass through by mapping the negative values to zero.

The output layer of the network gives out the predictions to which an activation function is applied to produce probability estimations in classification problems. In binary classification problems, a single neuron in the output layer is passed through sigmoid function. In multi-class problems, the output layer consists of multiple neurons each representing a specific class and softmax activation function is applied to produce the probability estimates for each class. The basic network diagram of a multi-layer perceptron with one hidden layer is shown in Fig. 1. The hidden and output layer nodes are calculated as

$$z_h = sigmoid(w_h^T x) = \frac{1}{1+exp[-(\sum_{j=1}^{d} w_{hj}x_j + w_{h0})]}$$
$$y_i = v_i^T z = \sum_{h=1}^{H} v_{ih}z_h + z_h + v_{i0} \tag{14}$$

In MLP, the backpropagation learning method, which is a type of stochastic descent method (Rumelhart et al., 1986), is used to train the network. The hidden layer weights, w, and output layer weights, v,

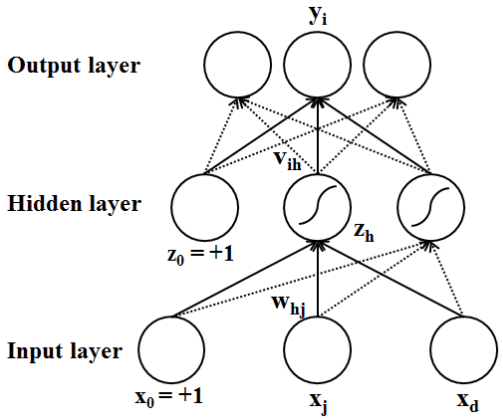

Figure 1: Multilayer perceptron architecture.

are updated according to the following rules until convergence:

$$\Delta v_h = \sum_t (r^t - y^t) z_h^t$$
$$\Delta w_{hj} = \eta \sum_t (r^t - y^t) v_h z_h^t (1 - z_h^t) x_j^t \tag{15}$$

3.4 Linear D-AR Method

To address the problems of CCA highlighted in introduction section, a two-view feature extraction method that aims to discover correlated and also discriminative linear features by utilizing class labels in the framework has already been proposed in (Sakar and Kursun, 2017). In linear D-AR, both views have their own MLPs where the input layer is composed of the their own view features. With the help of the hidden layer, input layer of each MLP-based D-ARNet is transformed into a lower dimensional subspace, then the hidden layer is mapped to the output layer which consists of both class labels and covariate outputs. Covariate outputs are alternated between the corresponding MLPs in order to maximize the correlation between two views. Having class labels in the output layer ensures to maximize the classification accuracy as well, while maximizing the correlation with covariate outputs. Class labels are not alternated between views and original class labels are used in each iteration. Training process of the network stops when the correlation of the outputs between two views do not change or iteration exceeds a certain limit.

The AR process starts with the first D-AR network of view 1. Correlated outputs, 's^x', hidden layer weights, 'w^x', and output layer weights, 'v^x' are initialized with random values. Then, training process starts for the first MLP with the given features, X, in the input layer. Hidden layer values, z^x, weights w^x and v^x are updated for the first network during

the training process and the final covariate outputs of view 1 's^x' are calculated.

The total error function of the first D-AR Network X can be written as

$$E_x(w^x, v^x|X) = E_s^x + \lambda E_r^x \tag{16}$$

where E_s^x and E_r^x are the errors of correlated output nodes and class label, respectively, λ is the discrimination factor which is used to trade off between the correlation of the output units and the discriminative ability of the network.

Since prediction of the correlated outputs is a regression problem, sum-of-squares error function is used to compute E_s^x. On the other hand, depending on the output variable type E_r^x is calculated differently. For classification problems like the emotion recognition task in this paper, cross-entropy function is used to compute E_r^x. For regression issues in which the output is a numerical value, sum-of-squares error function is used. Thus, the total error function in (16) can be re-written for classification problem as

$$\begin{aligned} E_x(w^x, v^x|X) &= E_s^x + \lambda E_r^x \\ &= \frac{1}{2}\Big(\sum_{t=1}^{N}\sum_{i=1}^{k}(s_{it}^y - s_{it}^x)^2\Big) \\ &\quad - \lambda\sum_{t=1}^{N}\sum_{i=1}^{p}(l_{it} log r_{it}^x) \end{aligned} \tag{17}$$

and for regression as

$$\begin{aligned} E_x(w^x, v^x|X) &= E_s^x + \lambda E_r^x \\ &= \frac{1}{2}\Big(\sum_{t=1}^{N}\sum_{i=1}^{k}(s_{it}^y - s_{it}^x)^2\Big) \\ &\quad + \frac{1}{2}\Big(\sum_{t=1}^{N}(r_t^{x'} - r_t^x)^2\Big) \end{aligned} \tag{18}$$

where N is the total number of instances, k is the number of hidden layer nodes which represents the number of the features extracted, 'w^x' and 'v^x' are the hidden layer weights and output layer weights respectively, l_{it} is 1 if sample x^t belongs to class i and 0 if not, and r_{it}^x is the predicted value of the ith class for the sample t, $r_t^{x'}$ is the predicted value of the tth sample in regression problem, s_{it}^x is the ith output of sample t for View 1, X.

Hidden layer weights, w^x, and output layer weights, v^x, of the MLPs are updated according to the back-propagation algorithm (Rumelhart et al., 1986).

$$\begin{aligned} \frac{\partial E^x}{\partial w_{hj}^x} &= \sum_{i=1}^{k}\frac{\partial E_s^x}{\partial s_i^x}\frac{\partial s_i^x}{\partial z_h^x}\frac{\partial z_h^x}{\partial w_{hj}^x} \\ &\quad + \lambda\sum_{i=1}^{p}\frac{\partial E_r^x}{\partial r_i^x}\frac{\partial r_i^x}{\partial z_h^x}\frac{\partial z_h^x}{\partial w_{hj}^x} \end{aligned} \tag{19}$$

where w_{hj}^x is the hidden layer weight between jth input node and the hth hidden layer node of view 1, and z_h^x is the hth hidden node of view 1. The correlated output units and predicted class labels of a given instance x_t are computed as follows:

$$s_{it}^x = \sum_{h=1}^{k} v_{ih}^x z_{ht}^x + v_{i0}^x r_{it}^x = \frac{exp\big(\sum_{h=1}^{k} v_{ih}^x z_{ht}^x + v_{i0}^x\big)}{\sum_{j=1}^{p} exp\big(\sum_{h=1}^{k} v_{jh}^x z_{ht}^x + v_{j0}^x\big)} \tag{20}$$

where v_{ih}^x is the output layer weight between hth hidden and the ith correlated output node of view 1. As it is seen in equation 24, the predicted values of the class outputs, r_{it}, are passed through softmax activation function in the output layer to obtain the probability estimates for each class. Output layer weights are shared by the class label and correlated output nodes with the aim of extracting discriminative features while maintaining the correlated information of the other view by producing the same outputs.

$$\begin{aligned} \Delta w_{hj}^x &= \eta_1 \sum_{t=1}^{N}\Big[\sum_{i=1}^{k}(s_{it}^y - s_{it}^x)v_{ih}^x\Big]x_{jt} \\ &\quad + \lambda\eta_2 \sum_{t=1}^{N}\Big[\sum_{i=1}^{k}(l_{it} - r_{it}^x)v_{ih}^x\Big]x_{jt} \\ \Delta v_{ih}^x &= \eta_1 \sum_{t=1}^{N}(s_{it}^y - s_{it}^x)z_{ht}^x \\ &\quad + \lambda\eta_2 \sum_{t=1}^{N}(l_{it} - r_{it}^x)z_{ht}^x \end{aligned} \tag{21}$$

where η_1 and η_2 are the learning factors of the covariate output and class labels respectively.

Same process applies for the second D-AR network of view 2, however, this time the covariate outputs of view2, 's^y', are not initialized randomly. Instead, covariate outputs of view 1 's^x' are fed into 's^y', while keeping the class labels fixed in the output layer. Once the training is completed for view 2, hidden layer values 'z^y', weights 'w^y' and 'v^y' are updated for the second network and the final set of covariate outputs of view 2 's^y' are calculated. This time s^y are fed into first view outputs, 's^x'. This iterative approach continues till the correlation of the outputs between two views do not change or iteration exceeds a certain limit.

One key note to highlight, as the alternated outputs tend to tune to the same direction to decrease the minimum square error, they need to be decorrelated before being fed into the other D-AR network. For this purpose, the cascading anti-Hebbian inhibition algorithm is used (Sakar and Kursun, 2017). The

inhibition rule after each epoch is:

$$s_{it}^{x} = s_{it}^{x} - \sum_{j=1}^{i-1} \rho(s_i^x, s_j^x) s_{jt}^{x} \tag{22}$$

where $\rho(s_i^x, s_j^x)$ is the correlation coefficient between ith and jth outputs of view 1. If 's_i^x' and 's_j^x' are extremely correlated with each other, then the ith output of view 1 is almost cleared.

4 PROPOSED NONLINEAR D-AR METHOD

Even though the linear D-AR method avoids the use of covariance matrices which are sensitive to outliers, similar to CCA, D-AR is limited to exploring only the linear relationships and cannot explore complex representations. The method proposed in this paper is the nonlinear version of (Sakar and Kursun, 2017). The architecture of the proposed nonlinear D-AR method is based on D-AR (Sakar and Kursun, 2017) and AR (Sakar et al., 2014b) methods, implemented by two multilayer perceptrons with nonlinear hidden layers. The network diagram of the non-linear D-AR method on multiple-output regression task is shown in Fig. 2. Using nonlinear activation function in the hidden layer helps to explore complex relationships from the views.

In the non-linear D-AR, different from linear D-AR, the hidden layer values are passed through a nonlinear activation function as:

$$z_{ht}^{x} = g\Big(\sum_{i=1}^{n} x_i^t w_{ih}^{x} + w_{i0}^{x}\Big) \tag{23}$$

where g is a nonlinear activation function such as sigmoid, hyperbolic tangent or rectified linear unit, and n is the number of features in view X. Update rules of the hidden and output layer weights are derived using gradient descent according to the activation function used in the hidden layer.

In this paper, we also propose to use the D-AR network for multiple-output regression problem. Thus, we aim to extract correlated features which carry predictive information about multiple numerical outputs. For multiple-output regression problem total error function can be re-written as:

$$\begin{aligned} E_x(w^x, v^x|X) &= E_s^x + \lambda E_r^x \\ &= \frac{1}{2}\Big(\sum_{t=1}^{N}\sum_{i=1}^{k}(s_{it}^{y} - s_{it}^{x})^2\Big) \\ &+ \lambda\frac{1}{2}\Big(\sum_{t=1}^{N}\sum_{i=1}^{m}(r_{ti}^{x'} - r_{ti}^{x})^2\Big) \end{aligned} \tag{24}$$

where m is the number of the outputs in the regression task. A single D-AR network is trained to minimize the total error on the multiple outputs. Thus, we aim to obtain a single set of features from each view that contain important predictive information about the target variables.

5 EXPERIMENTAL RESULTS

In our experiments, we have compared the discriminative power of our proposed nonlinear D-AR algorithm with linear D-AR, CCA and KCCA on the Cohn-Kanade (CK+) facial expression recognition dataset (Lucey et al., 2010) for classification and on the Residential Building dataset (Rafiei and Adeli, 2015) for 2-output regression. We use two different versions of nonlinear D-AR with sigmoid and Tanh nonlinear activation functions in the hidden layer. For evaluating the discriminative power of the features extracted with the methods used in this study, we use random forest (RF) algorithm for both classification and regression. For linear and nonlinear D-AR networks, the features extracted in the hidden layer of the networks are fed to RF (Breiman, 2001). For CCA and KCCA methods, the canonical variates are fed into RF. The number of ensemble trees in RF algorithm is selected as 100. Experiments are repeated for different training set sizes and the number of covariate outputs. For classification dataset, training sets are selected as 35 (5 instance from each of the 7 classes), 70 (10 instance from each of the 7 classes) and 105 (15 instance from each of the 7 classes). For regression dataset, training sets are selected as 35 and 70. The training and test data splits are repeated 10 times and for statistical significance. For classification the average of the accuracies and for regression the average of the total mean absolute errors obtained on 2-outputs (MAE) in these runs are reported.

For both linear and nonlinear D-AR networks, we have selected different number of covariate outputs, 1 to 7, for our experiments. The hidden layer of the networks contain 2 neurons in addition to the ones representing covariate outputs, 3 to 9. For CCA and KCCA, the number of covariate components are selected from 3 to 9, which is the same with the number of hidden layer nodes in D-AR network. Principal component analysis (LII, 1901) is applied before CCA and KCCA algorithms in order to improve the robustness of the methods. We should note that in our experiments the hidden layer is designed as a bottleneck layer in which the number of neurons is less than that of the output layer. This can be seen as an implicit regularization that enforces the networks tune to

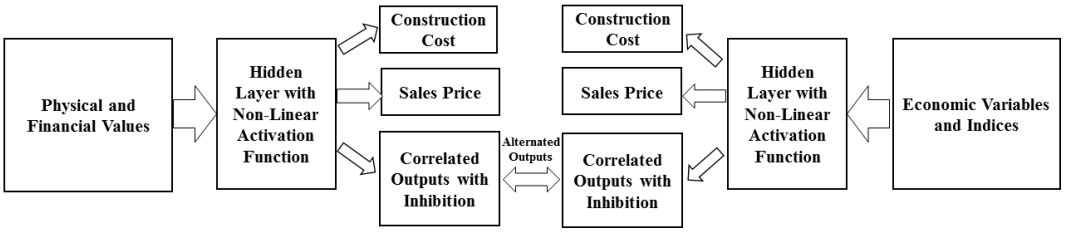

Figure 2: Block diagram of the non-linear version of D-AR on the multiple-output regression task: residential building dataset.

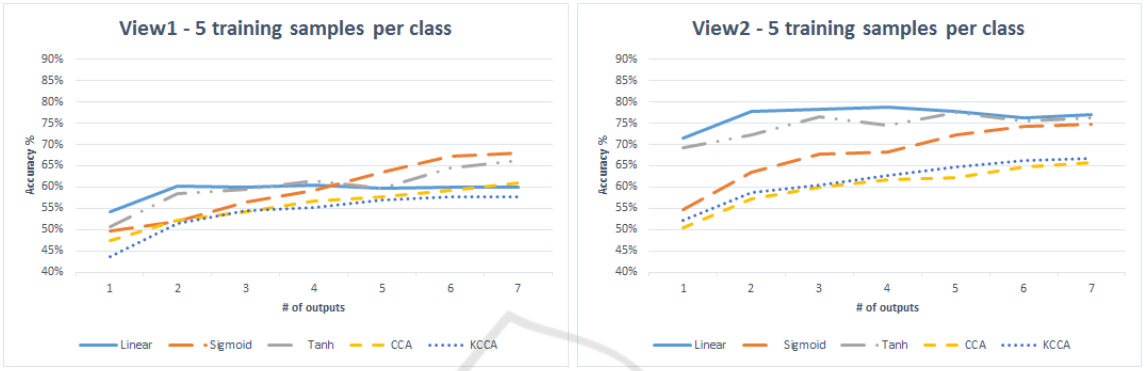

Figure 3: Cohn-Kanade (CK+) dataset: Number of covariate outputs versus accuracies obtained using 5 samples from each class. (left) Accuracy of the covariates extracted from View 1 (right) Accuracy of the covariates extracted from View 2.

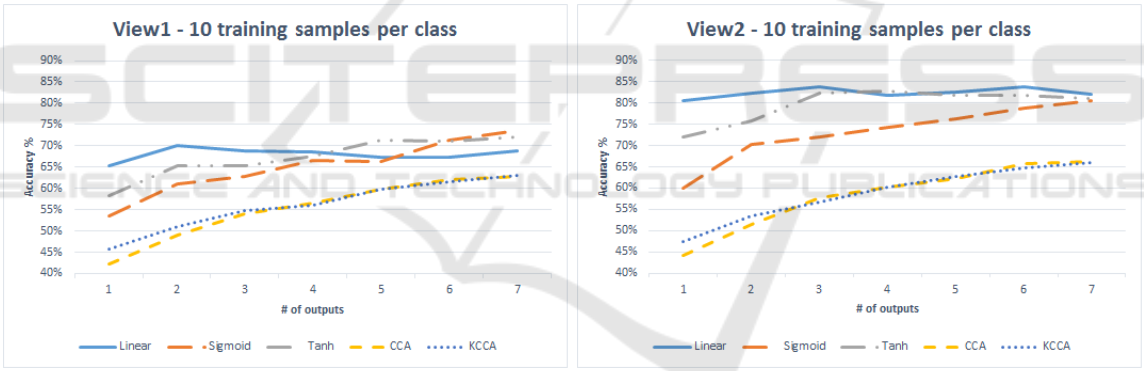

Figure 4: Cohn-Kanade (CK+) dataset:Number of covariate outputs versus accuracies obtained by using 10 samples from each class. (left) Accuracy of the covariates extracted from View 1 (right) Accuracy of the covariates extracted from View 2.

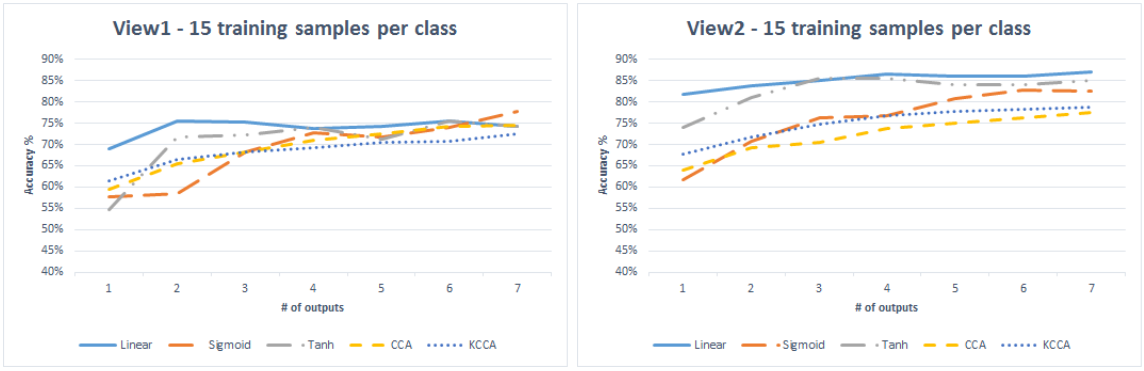

Figure 5: Cohn-Kanade (CK+) dataset:Number of covariate outputs versus accuracies obtained using 15 samples from each class. (left) Accuracy accuracy of the covariates extracted from View 1 (right) Accuracy of the covariates extracted from View 2.

Table 1: Cohn-Kanade (CK+) dataset:Covariate output correlations of View 1 and View 2 for training set.

Method	Output 1	Output 2	Output 3	Output 4	Output 5	Output 6	Output 7
CCA	100	100	100	100	100	100	100
KCCA	100	100	100	100	100	100	100
Linear D-AR	99	99	99	99	99	98	99
Sigmoid D-AR	90	89	40	35	29	16	18
Tanh D-AR	92	91	89	84	1	38	51

Table 2: Cohn-Kanade (CK+) dataset:Covariate output correlations of View 1 and View 2 for test set.

Method	Output 1	Output 2	Output 3	Output 4	Output 5	Output 6	Output 7
CCA	52	39	47	41	44	31	30
KCCA	67	46	45	43	32	34	37
Linear D-AR	80	62	62	48	48	43	11
Sigmoid D-AR	82	81	24	30	21	18	14
Tanh D-AR	79	76	62	45	11	23	30

most generalizable information at the expense of losing some rare relations which might be due to outliers in some cases.

5.1 Cohn-Kanade (CK+) Dataset

Fig. 3 shows the test set accuracies versus the number of covariate outputs obtained using 5 samples from each class. While the left chart displays the accuracy when the covariates extracted from view 1 are fed to RF algorithm, the right chart displays the accuracies obtained with view 2 covariates. Figures 4 and 5 display the accuracies when training set is selected as 10 and 15 samples from each class, respectively. In general, it is seen that view 2 has better classification accuracy when compared to view 1 for all methods and training set sizes. Thus, we can conclude that the discriminative power of the features extracted from view 2 are higher than those extracted from view 1.

As it can be seen from the figures, the accuracy obtained with the features of linear D-AR network surpasses CCA which is in parallel to the results obtained with different classifiers in (Sakar and Kursun, 2017). We also see that the accuracies obtained with the features of both versions of nonlinear D-AR network, sigmoid and tanh, are higher than that of KCCA. In figure 5, it is seen that when we have sufficient number of classes from each set (15), the discriminative performances of the methods are getting closer to each other when compared to figures 3 and 4. On the other hand, when we have limited information for each view, D-AR networks learn more from each other and gain more advantage over CCA and KCCA. During training phase of D-AR networks, both views interact and learn from each other and further improve their own discriminative accuracy using correlated outputs and class labels together.

With the increase in the number of covariate outputs, the accuracy first increases, stabilizes after some point and then fluctuates. Another important observation is that the accuracy of the nonlinear D-AR with sigmoid function increases more with increasing number of covariate outputs when compared to its linear version. This is because the linear D-AR is limited to explore linear relationships and cannot explore additional complex relations with limited training sample size. We should also note that although in general linear D-AR provides the highest accuracy for view 2, it does not improve the accuracy of the other view significantly. On the other hand, with increasing number of covariates, the nonlinear D-AR improves the performance of both view 1 and view 2 which shows that the networks guide each other well during the alternating regression procedure. As a result of this interaction, in view 2, nonlinear D-AR achieves the performance of linear D-AR with more covariate outputs, and in view 1, the features of nonlinear D-AR surpass the features of linear D-AR in classification performance.

Table 1 and 2 display the covariate correlations of View 1 and View 2 explored by CCA and KCCA along with the correlations of the covariate outputs of the D-AR networks for the training and test sets, respectively. The training set correlations of CCA and KCCA presented in these tables show that these methods overfit to the training set and do not generalize well on the test set. All 3 versions of the D-AR networks have higher correlations on the test set than CCA and KCCA. These results are in parallel with the accuracies obtained on the emotion recognition task.

5.2 Residential Building Dataset

As we have two outputs in this dataset, the results are computed and shown in terms of the sum of MAEs on output 1 and output 2. Fig. 6 shows the sum of the

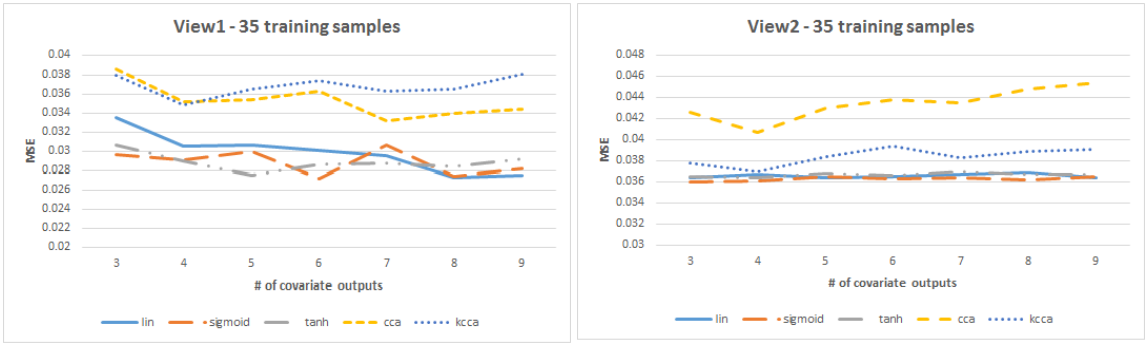

Figure 6: Residential Building dataset: Number of covariate outputs versus MSE obtained using 35 training samples. (left) MSE of the covariates extracted from View 1 (right) MSE of the covariates extracted from View 2.

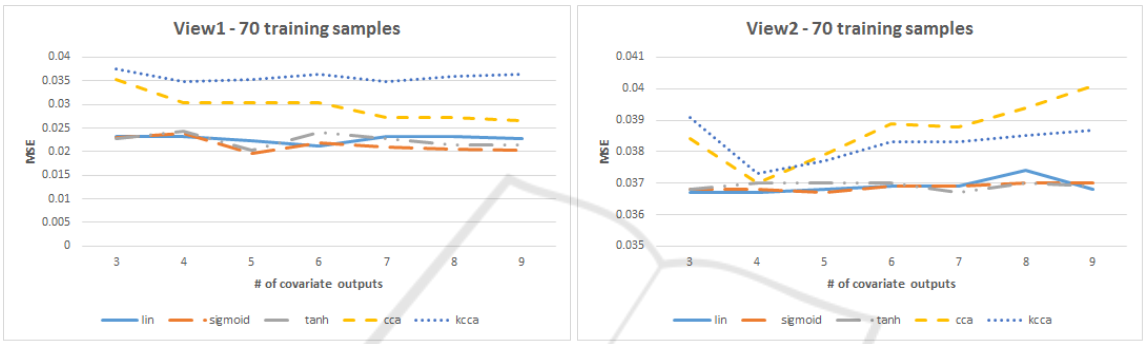

Figure 7: Residential Building dataset:Number of covariate outputs versus MSE obtained by using 70 training samples. (left) MSE of the covariates extracted from View 1 (right) MSE of the covariates extracted from View 2.

mean absolute errors (MAE) obtained on 2-outputs versus the number of covariate outputs using 35 training samples. While the left chart displays the sum of MAE when the covariates extracted from view 1 are fed to RF algorithm, the right chart displays the sum of MAE obtained with view 2 covariates. Fig. 7 displays the sum of MAE when 70 training samples are used for training.

As it can be seen from the figures, the MAE obtained with CCA features is higher than that of linear D-AR which is in parallel to the classification results of the Cohn-Kanade (CK+) dataset. We also see that the MAE calculated with the features of both versions of nonlinear D-AR network, sigmoid and tanh, are less than that of KCCA. As seen in Figure 6, nonlinear and linear versions of D-AR network of View 2 are very similar for all values of covariate outputs. On the other hand, for lower values of covariate outputs the features extracted from View 1 using linear D-AR network has higher error rate when compared to the non-linear versions.

In figure 7, it is seen that when we have sufficient number of training samples, performances of the methods are getting closer to each other when compared to figure 6. Similar to classification experiment, when we have limited information for each view, D-AR networks learn more from each other and gain more advantage over CCA and KCCA. During training phase of D-AR networks, both views interact and learn from each other and further improve their own performance. With the increase in the number of covariate outputs, the MAE first decreases, stabilizes after some point and then fluctuates.

6 CONCLUSIONS

Kernel canonical correlation analysis (KCCA) aims to find the nonlinear relationships between two multidimensional views that are related with each other. Although KCCA features can be used for classification and regression problems, KCCA tends to overfit to the training set without proper regularization. Besides, KCCA is an unsupervised technique which does not utilize class labels or numerical target variables for feature extraction.

In this paper, we propose the nonlinear version of the discriminative alternating regression (D-AR) method which uses target information during feature extraction. The nonlinear D-AR combines two alternating multilayer perceptrons (MLP) with nonlinear

hidden layers. We also propose to use D-AR network for multiple-output regression task. The discriminative and predictive performance of the features extracted with the proposed nonlinear D-AR network is compared to that of linear D-AR, CCA and KCCA algorithms. We use random forest algorithm as the base classifier. Experimental results on publicly available emotion recognition and residential building dataset show that the features of the nonlinear D-AR network give significantly higher accuracies and less errors than that of KCCA on classification and regression problems, respectively. Another important finding is that although KCCA explores highly correlated covariates on the training set, all versions of the D-AR network have higher correlations on the test set than CCA and KCCA, which is in parallel with the test set performances obtained on the supervised learning tasks.

As a future research direction, advanced regularization techniques can be applied to both KCCA and the proposed network to improve their robustness against outliers. The robustness of KCCA can be improved using a reduced kernel method while the proposed method can be improved using weight decay mechanism or another backpropagation algorithm such as resilient backpropagation with weight backtracking.

ACKNOWLEDGEMENTS

This research has been supported by Turkish Scientific and Technological Research Council (TUBITAK) project 215E008.

REFERENCES

Akaho, S. (2001). A kernel method for canonical correlation analysis. In *In Proceedings of the International Meeting of the Psychometric Society (IMPS2001*. Springer-Verlag.

Asuncion, A. and Newman, D. (2007). Uci machine learning repository. irvine, ca: University of california, school of information and computer science. *URL [http://www. ics. uci. edu/_ mlearn/MLRepository. html]*.

Bach, F. R. and Jordan, M. I. (2003). Kernel independent component analysis. *J. Mach. Learn. Res.*, 3:1–48.

Biemann, F., Meinecke, F. C., Gretton, A., Rauch, A., Rainer, G., Logothetis, N. K., and Mller, K. R. (2010). Temporal kernel cca and its application in multimodal neuronal data analysis. 79.

Branco, J. A., Croux, C., Filzmoser, P., and Oliveira, M. R. (2005). Robust canonical correlations: A comparative study. *Computational Statistics*, 20(2):203–229.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Cai, J. and Huang, X. (2017). Robust kernel canonical correlation analysis with applications to information retrieval. *Eng. Appl. Artif. Intell.*, 64(C):33–42.

Chen, J., Bushman, F. D., Lewis, J. D., Wu, G. D., and Li, H. (2012). Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis. 14.

He, Y., Zhao, L., and Zou, C. (2005). Face recognition based on pca/kpca plus cca. In *Advances in Natural Computation*, pages 71–74, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hotelling, H. (1992). *Relations Between Two Sets of Variates*, pages 162–190. Springer New York, New York, NY.

Hsieh, W. W. (2000). Nonlinear canonical correlation analysis by neural networks. *Neural Netw., vol. 13, no. 10, pp. 10951105*.

Huang, S. Y., Lee, M. H., and Hsiao, C. K. (2009). Nonlinear measures of association with kernel canonical correlation analysis and applications. *Journal of Statistical Planning and Inference*, 139(7):2162 – 2174.

Karaali, A. (2012). Face detection and facial expression recognition using moment invariants.

Lai, P. L. and Fyfe, C. (1998). Canonical correlation analysis using artificial neural networks. *Proc. 6th Eur. Symp. Artif. Neural Netw., Bruges, Belgium, Apr, pp. 363367*.

Lee, Y. J. and Huang, S. Y. (2007). Reduced support vector machines: A statistical theory. *IEEE Transactions on Neural Networks*, 18(1):1–13.

Li, Y. and Shawe-Taylor, J. (2006). Using kcca for japanese—english cross-language information retrieval and document classification. *J. Intell. Inf. Syst.*, 27(2):117–133.

LII, K. P. F. S. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572. PCA beginnings.

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J. M., Ambadar, Z., and Matthews, I. A. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 94–101.

Melzer, T., Reiter, M., and Bischof, H. (2001). Nonlinear feature extraction using generalized canonical correlation analysis. In *Artificial Neural Networks — ICANN 2001*, pages 353–360, Berlin, Heidelberg. Springer Berlin Heidelberg.

Pezeshki, A., Azimi-Sadjadi, M. R., and Scharf, L. L. (2003). A network for recursive extraction of canonical coordinates. *Neural Netw., vol. 16, nos. 56, pp. 801808*.

Rafiei, M. and Adeli, H. (2015). Novel machine learning model for estimation of sale prices of real estate units. *ASCE, Journal of Construction Engineering & Management, 142(2), 04015066*.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Neurocomputing: Foundations of research. chapter Learning Internal Representations by Error Propagation, pages 673–695. MIT Press, Cambridge, MA, USA.

Sakar, C. O. and Kursun, O. (2017). Discriminative feature extraction by a neural implementation of canonical correlation analysis. *IEEE Transactions on neural networks and learning systems, Vol.28, No.1*.

Sakar, C. O., Kursun, O., and Gurgen, F. (2014a). Ensemble canonical correlation analysis. *Applied Intelligence*, 40(2):291–304.

Sakar, C. O., Kursun, O., and Gurgen, F. (2014b). Feature extraction based on discriminative alternating regression. In *PRoa Romero L. (eds) XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013. IFMBE Proceedings*, MEDICON'2013.

Sakar, C. O., Kursun, O., Karaali, A., and Erdem, C. E. (2012). Feature extraction for facial expression recognition by canonical correlation analysis. In *Proc. IEEE 20th Signal Process. Appl. Conf., Mugla, Turkey pp. 1-3*.

Schölkopf, B. (2000). The kernel trick for distances. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, pages 283–289, Cambridge, MA, USA. MIT Press.

Sun, T. and Chen, S. (2007). Locality preserving cca with applications to data visualization and pose estimation. 25:531–543.

Ulukaya, S. (2011). Affect recognition from facial expressions for humancomputer interaction. MSc Thesis, Bahcesehir University.

Wold, H. (1966). Nonlinear estimation by iterative least squares procedures. *Research Papers in Statistics (Festschrift for J. Neyman), F. N. David, Ed. New York, NY, USA: Wiley, pp. 411444*.

Yeh, Y. R., Huang, C. H., and Wang, Y. C. F. (2014). Heterogeneous domain adaptation and classification by exploiting the correlation subspace. *IEEE Transactions on Image Processing*, 23(5):2009–2018.

Zhu, X., Huang, Z., Shen, H. T., Cheng, J., and Xu, C. (2012). Dimensionality reduction by mixed kernel canonical correlation analysis. *Pattern Recogn.*, 45(8):3003–3016.